Adapting ASPEN for Orbital Express

Declarative modeling brings efficiency to encoded procedures and allows for guarantees on resource usage and time usage.

NASA’s Jet Propulsion Laboratory, Pasadena, California

By studying the Orbital Express mission, modeling the spacecraft and scenarios, and testing the system, a technique has been developed that uses recursive decomposition to represent procedural actions declaratively, schema-level uncertainty reasoning to make uncertainty reasoning tractable, and lightweight natural language processing to automatically parse procedures to produce declarative models.

Schema-level uncertainty reasoning has, at its core, the basic assumption that certain variables are uncertain, but not independent. Once any are known, then the others become known. This is important where a variable is uncertain for an action and many actions of the same type exist in the plan. For example, if the number of retries to purge pump lines was unknown (but bounded), and each attempt required a sub-plan, then, once the correct number of attempts required for a purge was known, it would likely be the same for all subsequent purges. This greatly reduces the space of plans that needs to be searched to ensure that all executions are feasible.

To accommodate changing scenario procedures, each is ingested into a tabular format in temporal order, and a simple natural-language parser is used to read each step and to derive the impact of that step on memory, power, and communications. Then an ASPEN (Activity Scheduling and Planning Environment) model is produced based on this analysis. The model is tested and further changed by hand, if necessary, to reflect the actual procedure. This results in a great savings of time used for modeling procedures.

Many processes that need to be modeled in ASPEN (a declarative system) are, in fact, procedural. ASPEN includes the ability to model activities in a hierarchical fashion, but this representation breaks down if there is a practically unbounded number of sub-activities and decomposition topologies. However, if recursive decomposition is allowed, HTN-like encodings are enabled to represent most procedural phenomena.

For example, if a switch requires a variable (but known at the time of the attempt) number of attempts to switch on, one can recurse on the number of remaining switch attempts and decompose into either the same switching activity with one less required attempt, or not decompose at all (or decompose into a dummy task), resulting in the end of the decomposition. In fact, any bounded procedural behavior can be modeled using recursive decompositions assuming that the variables impinging the dis-
junctive decomposition decision are computable at the time that the decision is made. This enables one to represent tasks that are controlled outside of the scheduler, but that the scheduler must accommodate, without requiring one to give a declarative model of the procedural behavior. 

This work was done by Caroline Chouinard, Daniel Tran, Grailing Jones, Van Dang, and Russell Knight of Caltech for NASA’s Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Daniel Broderick of the California Institute of Technology at danielb@caltech.edu. Refer to NPO-45262.